

质量属性分类			描述	评分策略	分值
软件质量 (外部质量)	功能		是指当软件在指定条件下使用，软件产品满足明确和隐含要求功能的能力，包含以下子特性： 适合性：软件产品为指定的任务和用户目标提供一组合适功能的能力。 准确性：软件提供给用户功能的精确度是否符合目标。 互操作性：软件与其它系统进行交互的能力。 保密安全性：软件保护信息和数据的安全能力。	以功能实现bug数量为度量依据，分为致命问题、严重问题和一般问题三个bug级别，每出现一个致命问题减5分，每出现一个严重问题减3分，每出现一个一般问题减2分，减完为止。	30
	性能		是指在规定条件下，相对于所用资源的数量，软件产品可提供适当的性能的能力。主要包含时间性能和空间性能。	对于非性能相关代码或者无需关注性能，可以给性能基准分6分。 对于性能相关代码，基本能够满足性能指标要求，无明显性能问题，给10分。 对于性能相关代码，使用了巧妙的设计能够显著提升性能，可以大于10分，但最多不能超过20分。	10~20
	其他	可靠性	在指定条件下使用时，软件产品维持规定的性能水平的能力，包含以下子特性： 健壮性：软件产品为避免软件内部的错误扩散而导致系统失效的能力。 容错性：软件防止外部接口错误扩散而导致系统失效的能力。 易恢复性：系统失效后，重新恢复原有的功能和性能的能力。	其他非功能质量属性，当前只作为加分项，暂不纳入必评范围，评审委员可以基于这几个维度去挖掘开发人员代码中的设计和实现亮点，酌情给分，最多不超过10分。	10~20
		可维护性	是指软件产品可被修改的能力，修改可能包括修正，改进或软件适应环境、需求和功能规格说明中的变化。 易分析性：软件提供辅助手段帮助开发人员定位缺陷产生的原因，判断出修改的地方。 易改变性：软件产品使得指定的修改容易实现的能力。 稳定性：软件产品避免由于软件修改而造成意外结果的能力。 易测试性：软件提供辅助性手段帮助测试人员实玩其测试意图。		
		可移植性	是指软件产品从一种环境迁移到另一种环境的能力。 适应性：软件产品无需作相应变动就能适应不同环境的能力。 易安装性：尽可能少的提供选择，方便用户直接安装。 共存性：软件产品在公共环境中与其它软件分享公共资源共存的软件。 易替换性：软件产品在同样的环境下，替代另一个相同用途的软件产品的能力。		
代码质量 (内部质量)	代码设计		关注模块内部的设计是否合理。	主要依据设计模式中设计原则进行评判： 1. 单一职责原则 2. 里氏替换原则 3. 依赖倒置原则 4. 接口隔离原则 5. 迪米特法则 6. 开放封闭原则 总分20分，每发现一处违背设计原则的地方扣2分，扣完为止。	20
	代码实现	编程规范	代码与编码规范符合程度。	以公司发布的编码规范为依据，每发现一处与规范不一致的地方扣1分，同一类型的编码规范问题扣分不超过2分。对于不适合已发布的编码规范的情况，最高不能超过8分。	10
		代码坏味道	代码中是否有坏味道。	主要依据《重构-改善既有代码的设计》中列举的代码坏味道。 总分20分，每发现一处代码坏味道扣1分，重复类型坏味道会重复扣分，扣完为止。  章节三 代码的坏味道 ❶ Duplicated Code (重复的代码) ❷ Long Method (过长函数) ❸ Large Class (过大类) ❹ Long Parameter List (过长参数列表) ❺ Divergent Change (发散式变化) ❻ Shotgun Surgery (散弹式修改) ❼ Feature Envy (依恋情结) ❽ Data Clumps (数据抱团) ❾ Primitive Obsession (基本型别偏执) ❿ Switch Statements (switch语句现身) ⓫ Parallel Inheritance Hierarchies (平行继承体系) ⓬ Lazy Class (冗赘类) ⓭ Speculative Generality (弯弯绕绕未来性) ⓮ Temporary Field (令人迷惑的临时值域) ⓯ Message Chains (过度耦合的消息链) ⓰ Middle Man (中间转手人) ⓱ Inappropriate Intimacy (亲密关系) ⓲ Alternative Classes with Different Interfaces (异曲同工类) ⓳ Incomplete Library Class (不完善的程序库类) ⓴ Data Class (纯相的数据类) ⓵ Refused Bequest (被拒绝的继承) ⓶ Comments (过多的注释)	20